IN THE CLAIMS:

Please amend claims 1, 14, and 28, as indicated hereinbelow.

Claims 2, 8, 9, 15, 21, 22, 27, 29, 35 and 36 are cancelled, as indicated hereinbelow.

Claims 3-7, 10-13, 16-20, 23-26, 30-34 and 37-40 are dependent claims which remain *"as is"*.

1. (Currently Amended). A method for adapting a legacy software application that includes interface specification definitions which include definitions of screen formats, created from legacy source code including data fields and developed for an environment comprising a centralized computing resource interconnected to a series of computer terminal devices, to a network environment, wherein said network environment comprises a system of distributed, interconnected network computing resources, said method comprising the steps of:

providing a software application which utilizes said legacy source code to automatically ~~produce~~ generate a series of user interface software components from the screen format definitions to provide a graphical user interface providing at least data entry and display facilities of the interface specification definitions, the said graphical user interface being embodied in a series of executable software components executable by a scripting language running on said interconnected network computing resource that provide the functionality for interaction with the legacy software application, said components containing object oriented methods for setting or obtaining values of said data fields and being executable by at least one of said computing resources in said network environment, and wherein upon execution, said computing resource is caused to interconnect with said legacy software application over said network so as to interact with said legacy software application in the transmission or receipt of information to and from said legacy software application.

2. (Cancelled).

3. (Currently Amended). A method in accordance with Claim 1 [[2]], wherein the interface software components are arranged to generate forms corresponding to forms generated by the legacy software application.

4. (Previously Presented). A method in accordance with Claim 1, comprising the step of generating client interface components, the client interface components being arranged to interact over the network with the legacy software application.

5. (Currently Amended). A method in accordance with Claim 3 [[4]], the client interface components include a user input object which is arranged to receive data input by a user of the network computing resource and transmit the data to the legacy application, over the network.

6. (Previously Presented). A method as claimed in Claim 1, wherein said series of software components are loadable and executable by an Internet Browser.

7. (Previously Presented). A method as claimed in Claim 1, any previous claim wherein said series of software components comprise Java code applets.

8. (Cancelled).

9. (Cancelled).

10. (Previously Presented). A method as claimed in Claim 1, wherein said network environment comprises the Internet network.

11. (Previously Presented). A method as claimed in Claim 1, wherein said network environment utilizes TCP/IP transfer protocols.

12. (Previously Presented). A method as claimed in Claim 1, wherein said source code is written in a 4GL language.

13. (Original). A method as claimed in Claim 12 wherein said source code is written in the LINC language.

14. (Currently Amended). A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform method steps for adapting a legacy software application that includes interface specification definitions which include definitions of screen formats created from legacy source code including data fields and developed for an environment comprising a centralized computing resource interconnected to a series of computer terminal devices, to a network environment, wherein said network environment comprises a system of distributed, interconnected network computing resources, said method comprising the steps of:

providing a software application which utilizes said source code to automatically ~~produce~~ generate a series of user interface software components from the screen format definitions to provide a graphical user interface providing at least data entry and display facilities of the interface specification definitions, the said graphical user interface being embodied in a series of executable software components executable by a scripting language running on said interconnected network computing resource that provides the functionality for interaction with the legacy software application, said components containing objected oriented methods for setting and obtaining values of said data fields and being executable by at least one of said computing resources in said network environment, and wherein upon execution, said computing resource is caused to interconnect with said legacy software application over said network so as to interact with said legacy software application in the transmission or receipt of information to and from said legacy software application.

15. **(Cancelled).**

16. (Currently Amended). A program storage device in accordance with Claim 11 [[15]], wherein the interface software components are arranged to generate forms corresponding to forms generated by the legacy software application.

17. (Currently Amended). A program storage device in accordance with Claim 11 [[14]], the method further comprising the step of generating client interface components, the client interface components being arranged to interact over the network with the legacy software application.

18. (Original). A program storage device in accordance with Claim 17, wherein the client interface components include a user input object which is arranged to receive data input by a user of the network computing resource and transmit the data to the legacy application, over the network.

19. (Currently Amended). A program storage device in accordance with Claim 11 [[14]], wherein said series of software components are loadable and executable by an Internet Browser.

20. (Currently Amended). A program storage device in accordance with Claim 11 [[14]], wherein said series of software components comprise Java code applets.

21. (Cancelled).

22. (Cancelled).

23. (Currently Amended). A program storage device in accordance with Claim 11 [[14]], wherein said network environment comprises the Internet network.

24. (Currently Amended). A program storage device in accordance with Claim 11 [[14]], wherein said network environment utilized TCP/IP transport protocols.

25. (Currently Amended). A program storage device in accordance with Claim 11 [[14]], wherein said source code is written in the LINC language.

26. (Currently Amended). A program storage device in accordance with Claim 11 [[14]], wherein said terminal screen definitions are written in a screen control language.

27. (Cancelled).

28. (Currently Amended). A system for adapting a legacy software application that includes interface specification definitions which include definitions of screen formats, created from legacy source code including data fields and developed for an environment comprising a centralized computing resource interconnected to a series of computer terminal devices, to a network environment, wherein said network environment comprises a system of distributed, interconnected network computing resources, the system comprising:

means utilizing said legacy source code to automatically ~~produce~~ generate a series of user interface software components from the said screen format definitions to provide a graphical user interface providing at least data entry and display facilities of the interface specification definitions, the said graphical user interface being embodied in a series of executable software components executable by a scripting language running on said interconnected network computing resource for providing the functionality for interaction with the said legacy software application, said executable software components containing object oriented methods for obtaining values of said data fields and being executable by at least one of said computing resources in said network environment, and wherein upon execution, said computing resource is caused to interconnect with said legacy software application over said network so as to interact with said legacy software application in the transmission or receipt of information to and from said legacy software application.

29. (Cancelled).

30. (Currently Amended). A system in accordance with Claim 28 [[29]], wherein the interface software components are arranged to generate forms corresponding to forms generated by the legacy software application.

31. (Previously Presented). A system in accordance with Claim 28, the means for producing the series of software components including means for generating client interface components, the client interface components being arranged to interact over the network with the legacy software application.

32. (Previously Presented). A system in accordance with Claim 31, the client interface components including a user input object which is arranged to receive data input by a user of the network computing resource and transmit the data to the legacy application, over the network.

33. (Previously Presented) A system in accordance with Claim 28, wherein said series of software components are loadable and executable by an Internet Browser.

34. (Previously Presented). A system in accordance with Claim 28, wherein the series of software components comprise Java code applets.

35. (Cancelled).

36. (Cancelled).

37. (Original). A system in accordance with Claim 28, wherein said network environment comprises the Internet network.

38. (Original). A system in accordance with Claim 28, wherein said network environment utilizes TCP/IP transfer protocols.

39. (Original). A system in accordance with Claim 28, wherein said source code is written in at 4GL language.

40. (Original). A system in accordance with Claim 28, wherein said source code is written in the LINC language.